REMARKS

Claims 1-12 are remaining in this application. Claim 2 has been amended.

Corrected drawings are submitted herewith as required by the Examiner.

The abstract has been amended as required by the Examiner.

Claim 2 has been amended to remove informalities.

Claims 1-5, 8 and 11 stand rejected under 35 U.S.C. §102(a) as being anticipated by Marcelais et al. Applicants respectfully traverse this rejection because the cited reference does not disclose or suggest the features of the dynamic variable specifying means or the area specifying means described in claim 1.

More specifically, the information processing apparatus of the present invention is specifically for translating a source file including a dynamic variable, as opposed to static variables, and includes dynamic variable specifying means for specifying a target dynamic variable from the source file and area specifying means for specifying areas insured in the case of a dynamic variable specified by the dynamic variable specifying means being developed into a memory.

In programming languages using static variables, the allocation of a static variable to a storage area is constant regardless of the program being executed. On the other hand, allocation of a dynamic variable to a memory area is determined when a program is executed. In order to initialize the contents of a dynamic variable to a predetermined value, it is necessary to know the memory area to which the dynamic variable was allocated by the

operating system (OS). In the present invention, the dynamic variable specifying means specifies target dynamic variables from a source file and area specifying means specifies areas in which the dynamic variables are stored in the memory.

The Marcelais et al. reference relates to a system and method for pre-processing variable initializers. It does not even recognize the problem associated with an initialization of a dynamic variable, and accordingly, does not disclose or suggest the features of the dynamic variable specifying means or the area specifying means, as in the present invention. For these reasons, independent claims 1 and 11 which both recite these features are allowable over the cited reference. Claims 2-10 depend from claim 1 and are also allowable for the same reasons.

Claims 6, 7 9 and 10 stand rejected under 35 U.S.C. §103 (a) as being unpatentable over Marcelais et al. in view of other references of record. Applicants respectfully traverse this rejection. For the same reasons given with respect to independent claim 1, for which these claims depend.

Claim 12 stands rejected under 35 U.S.C. §103 (a) as being unpatentable over Marcelais et al. in view of Kolawa et al. Applicants respectfully traverse this rejection because the cited references, even if combined, still would not disclose or suggest the features of the array specifying means or the area ensuring means for ensuring areas in a memory being a predetermined number of bytes more than areas declared in an array specify by the array specifying means, as described in claim 12.

The Examiner recognizes that the Marcelais et al. reference does not disclose the issuing area ensuring means of the invention, but that this feature is disclosed in the Kolawa et al. reference. The Kolawa et al. reference discloses a method for automatically detecting an array operation that is attempting to access an invalid memory location, and teaches employing a parse tree which is instrumented with debugging functionality (see col. 9, lines 58-62). The reference further teaches that the criteria for inserting this type of error check requires the array variable and its size be declared to the error-checking engine and that for each write operation to that array, the error-checking engine must check if the index into the array is valid, (see col. 9, line 65-col. 10, line 2). Thus, the Kolawa et al. reference merely teaches detecting an array operation that attempts to access an invalid memory location.

It does not, however, disclose or suggest any means for ensuring areas in the memory which is a predetermined number of bytes more than areas declared in a specified array, as in the present invention. In other words, detecting an array operation that is attempting to access an invalid memory location is not the same as ensuring that there are more areas in the memory (by a predetermined number of bytes) than that which is declared in the specified array. Therefore, even if Kolawa et al. were combined with Marcelais et al. they still would not disclose or suggest at least the area ensuring means of the present invention. For this reason, claim 12 is allowable over the cited references.

Moreover, it would have not been obvious to combined the cited references in the first place because there is no suggestion or teaching to do so. The Marcelais et al. reference relates to the pre-processing of variable initializers to solve the latency problem in start-up time due to the initialization of variables in source code (see col. 1, lines 25-43). The Kolawa et al. reference, on the other hand, is directed to a method for automatically instrumenting a computer program for debugging. The portion of Kolawa et al. relied on by the Examiner in his rejection specifically relates to an error-checking engine for automatically detecting an array operation that is attempting to access an invalid memory location. The problems that the Kolawa et al. reference seeks to solve is unrelated to the problem confronted by Marcelais et al. Therefore, one of ordinary skill in the art working to solve the problems described in Marcelais et al. would not have looked to the teachings of Kolawa et al. to derive the present invention. Claim 12 is allowable for this reason also.

For all of the above reasons, Applicants request reconsideration and allowance of the claimed invention. The Examiner should contact Applicants' undersigned attorney if a telephone conference would expedite prosecution.

Respectfully submitted,

GREER, BURNS & CRAIN, LTD.

By 

B. Joe Kim
Registration No. 41,895

December 16, 2003

Suite 2500
300 South Wacker Drive
Chicago, Illinois 60606
(312) 360-0080

Customer No. 24978
K:\0828\64986\64986 Amd A.doc